The unsupervised free CAT for low resource languages

SOURCESENSE

Alberto Massidda

Know your speaker

- Alberto Massidda
  - 36 yrs old, started in 2008
  - master CS
  - ex-startupper
- During the day: SRE and ML @ Sourcesense
  - Kubernetes, monitoring, CI/CD, Cloud, Hadoop, Python, Jupyter, sklearn, TF
- At night: Metalhead

Who we are

SOURCESENSE

- Founded in 2001;
- Branches in Milan, Rome, Cosenza and London;
- Market leader in enterprise ready solutions based on Open Source tech;
- Expertise:
    - DevOps: Monitoring, Cloud and Containers
    - Integration: Mobile, Frontend, Backend
    - Data science: ML, BigData and many more...

Outline

- The need for a standard toolchain
- The MateCAT tool
- Problem with today's Machine Translation
- New ideas with older technology at the rescue: Monoses
- What we brought on the table

The need for a standard toolchain

Sounds snide, but there is only one way to do it: the industrial way.

The industry actors, the Language Service Providers (LSP) have long set onto a consolidated set of processes, technologies and file formats.

This presentation is about exposing battle tested technologies to the community, so that we can return back soon to what we love: hacking.

The standard toolchain

- Computer Aided Translation (CAT) tool:
  an editor that parses a bilingual file (a special envelope/container for an "in-translation file") and manipulates its strings into other languages.
  Provides, among other things:
  - Tag editor: manipulates untranslatable entities, like markup.
  - Format preservation: converts from original to bilingual format and back.
- Translation Memory: a database of past translations, to be recycled or adapted for the future documents.
- Machine Translation: a server that provides translations on the fly.

Some dominant standards worth knowing

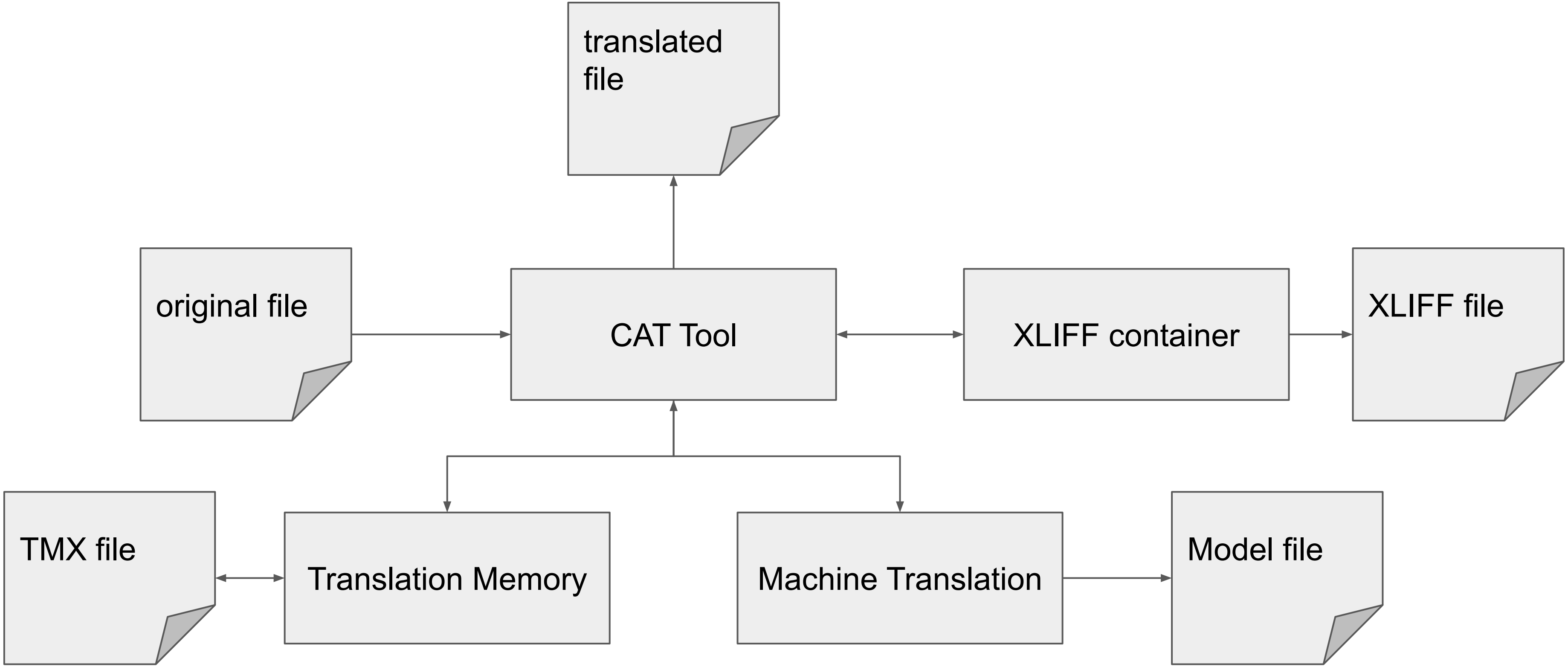
XLIFF: a XML based original file envelope that separates strings from markup.

TMX: a XML exchange format for translation memories.

PO: gettext localization strings format.

Strings/XML: iOS/Android localization strings format (how NOT to do it), use miracle2k/android2po and prop2po.

The standard workflow



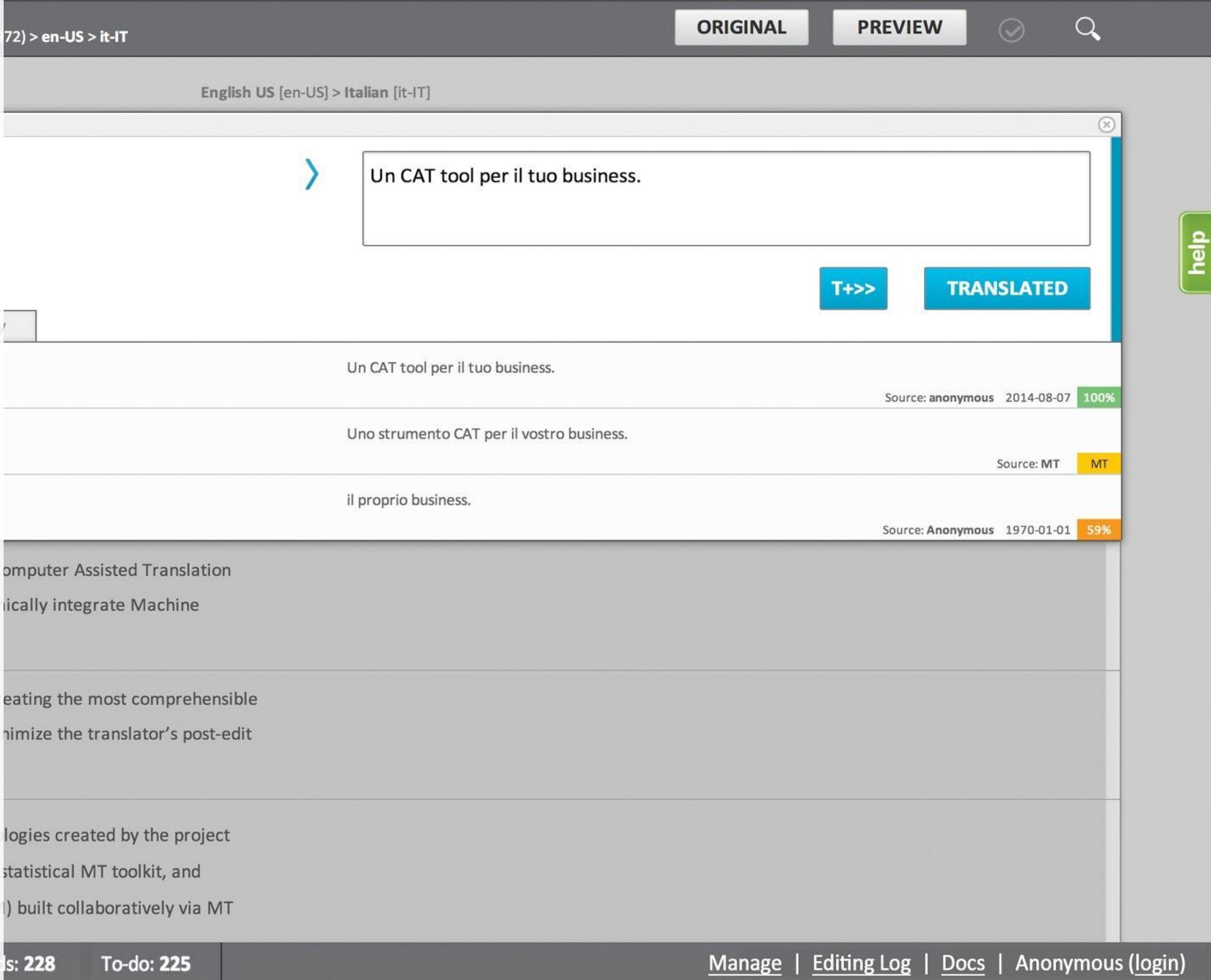
The MateCAT tool

An enterprise, FOSS, web based CAT tool.

Funded by EU in the 7th Framework programme, developed by Translated.

4 people team (including me, for the very first release).

[matecat/MateCat]()



English US [en-US] > Italian [it-IT]

Un CAT tool per il tuo business.

A CAT tool for your business. — Un CAT tool per il tuo business. — Source: anonymous 2014-08-07 100%

A CAT tool for your business. — Uno strumento CAT per il vostro business. — Source: MT MT

your business. — il proprio business. — Source: Anonymous 1970-01-01 59%

MateCat pushes what is considered the new frontier of Computer Assisted Translation (CAT) technology, that is, how to effectively and ergonomically integrate Machine Translation (MT) within the human translation workflow.

While current MT is mainly trained with the objective of creating the most comprehensible output, in MateCat we target MT technology that will minimize the translator's post-edit effort.

The project builds on state-of-the-art MT and CAT technologies created by the project members, such as Moses, the most popular open source statistical MT toolkit, and MyMemory, the world's largest Translation Memory (TM) built collaboratively via MT and human contributions.

Progress: 1% Payable Words: 228 To-do: 225 Manage | Editing Log | Docs | Anonymous (login)

Demo time

A critical part: the Machine Translation

No matter how much data we translate, we’ll never have enough memories.

A Machine Translation system gives a huge productivity boost.

MT are ML systems that are trained over datasets named “parallel corpus”.

Parallel corpus serve as a bidirectional labeled dataset.

Problem with today's Neural MT training

NMT are supervised models requiring labeled data. LOTS of labeled data.

Hundreds of millions of aligned words.

We barely have enough data for FIGS (French, Italian, German, Spanish).

Other languages have no chance to benefit from MT due to scarcity of resources.

Efforts of procuring more parallel data are undergoing.

OPUS: the open parallel corpus is an ever growing collection of parallel corpora.

The attractiveness of unsupervised training

Let’s take a step back.

We don’t need parallel examples to learn a language.

How about learning 2 languages and then try to map between concepts?

But, in order to map between languages, a computer needs to build a representation of that language. How can that be accomplished?

Word Embeddings and Language Models

Word Embedding is a technique to map every word into a vector space.

Words with “near meanings” will have “near vectors” in that space.

We can even do crazy things like:

$$v(\text{“}Paris\text{”}) - v(\text{“}France\text{”}) + v(\text{“}Italy\text{”}) = v(\text{“}Rome\text{”})$$

If we analyze enough sentences in one language, we start to develop a very structured model of how the language is built. A language model.

Building a parallel corpus out of air

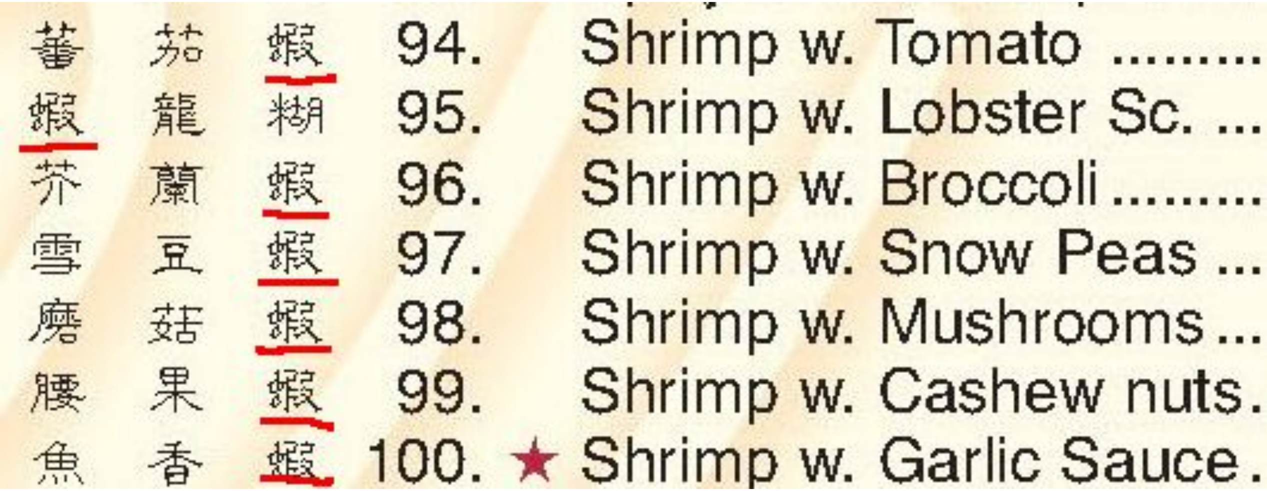
We could induce a parallel corpus between two independent languages by just mapping concepts between spaces (bootstrapping with a vocabulary or using unique entities and frequencies as heuristics).

Won’t the result be really noisy?

Sure, but we could use statistics to compute means and infer the true positives.

Phrase based Machine Translation

In good old fashioned days, we had statistical phrase-based MT: we counted co-occurrences of words and infer alignments between tokens.



蕃茄蝦	94.	Shrimp w. Tomato ....
蝦龍糊	95.	Shrimp w. Lobster Sc. ...
芥蘭蝦	96.	Shrimp w. Broccoli ....
雪豆蝦	97.	Shrimp w. Snow Peas ...
磨菇蝦	98.	Shrimp w. Mushrooms ...
腰果蝦	99.	Shrimp w. Cashew nuts.
魚香蝦	100. ★	Shrimp w. Garlic Sauce.

Statistics make translation probabilities

Co-occurrences counts between words and sequence of words (“phrases”) were used to calculate translation probabilities dataset-wise.

These probabilities were annotated in a database called “phrase table”.

Creating a phrase table is the most expensive operation of a PBMT.

The Moses toolchain

`IRSTLM` (and later, `KenML`) is the language model: calculates the probabilities of a sentences being meaningful.

`GIZA++` (and later, `fast_align`) calculate the phrase table.

`Moses` decodes the incoming message:

1. Projects the input sentence over the phrase table to retrieve translation options.
2. Searches among the different options guided by the LM as heuristic.
3. Stops when all input sentence has been covered.

Monoses

Artetxe, Labaka, Agirre. 2018. Unsupervised Statistical Machine Translation.

It is a toolkit to induce a phrase table from two monolingual datasets through word embedding mapping.

Computes a first model, does some fine tuning and then iteratively augments the original dataset by backtranslating the two monolingual corpora.

Noisy? Sure, but sheer amount of data averages out the noise in the long run.

Demo time

What we have today at FOSDEM

1. A Docker packaging of Monoses ready to use to generate a training set.
2. A HTTP API server to query the model obtained in this way.
   aijanai/monoses-server

3. A packaged version of Matecat tool, which includes MySQL, AMQ, daemons and Apache2 webapp. Runs on docker-compose, aims to support K8s.
   aijanai/docker_matecat

Kudos

- Philipp Koehn, for inventing the phrase based MT
- Thomas Mikolov, for inventing Word2Vec
- Adam Paszke, for inventing PyTorch
- Mikel Artetxe, for putting all together

Thanks